

This Page Is Inserted by IFW Operations  
and is not a part of the Official Record

## **BEST AVAILABLE IMAGES**

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images may include (but are not limited to):

- BLACK BORDERS
- TEXT CUT OFF AT TOP, BOTTOM OR SIDES
- FADED TEXT
- ILLEGIBLE TEXT
- SKEWED/SLANTED IMAGES
- COLORED PHOTOS
- BLACK OR VERY BLACK AND WHITE DARK PHOTOS
- GRAY SCALE DOCUMENTS

## **IMAGES ARE BEST AVAILABLE COPY.**

**As rescanning documents *will not* correct images,  
please do not report the images to the  
Image Problem Mailbox.**



GB9-2000-0083-US1

PATENT

- 1 -

BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

In re Application of:  
Michael C. Bailey et al.

: Before the Examiner:  
Shah, Nilesh R.

Serial No.: 09/628,367

: Group Art Unit: 2127

Filed: July 31, 2000

:

MAY 26 2004

Technology Center 2100

Title: METHOD, PROGRAM PRODUCT  
AND COMPUTER SYSTEM FOR  
PROGRESSIVE IMPROVEMENT  
OF AN ENVIRONMENT POOL

: IBM Corporation  
: P.O. Box 12195  
: Dept. T81/503  
: Research Triangle Park, NC 27709

**APPEAL BRIEF**

Mail Stop Appeal Brief-Patents  
Commissioner for Patents  
P.O. Box 1450  
Alexandria, VA 22313-1450

I. REAL PARTY IN INTEREST

The real party in interest is International Business Machines Corporation, which is the assignee of the entire right, title and interest in the above-identified patent application.

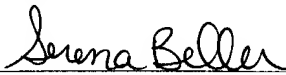
---

**CERTIFICATION UNDER 37 C.F.R. § 1.8**

I hereby certify that this correspondence is being deposited with the United States Postal Service with sufficient postage as first class mail in an envelope addressed to Mail Stop Appeal Brief-Patents, Commissioner for Patents, P.O. Box 1450, Alexandria, Virginia 22313-1450, on May 19, 2004.

05/25/2004 DEHHANU1 00000106 090461 09628367

01 FC:1402 330.00 DA

  
Signature

Serena Beller  
(Printed name of person certifying)

## II. RELATED APPEALS AND INTERFERENCES

There are no other appeals or interferences known to Appellants, Appellants' legal representative or assignee which will directly affect or be directly affected by or have a bearing on the Board's decision in the pending appeal.

## III. STATUS OF CLAIMS

Claims 1-18 are pending in the Application. Claims 1-18 stand rejected.

## IV. STATUS OF AMENDMENTS

The Appellants' response to the Office Action having a mailing date of September 2, 2003, has been considered, but the Examiner indicated that it did not place the application in condition for allowance because Appellants' arguments were deemed unpersuasive.

## V. SUMMARY OF INVENTION

In computer systems, it is well known in the art to have multiple programs and instances of the same program running at the same time. Specification, page 1, lines 13-15. Indeed, many of the problems and difficulties of managing modern systems have come about precisely because the information technology world has advanced since the days of dedicated batch processing systems in which only a single computing task could be executed at any given time. Specification, page 1, lines 15-20.

One problem that is particularly acute in modern systems is the impact on performance of initializing runtime environments. Specification, page 1, lines 22-24.

Various approaches have been taken to provide a solution to this problem. Specification, page 1, lines 26-27. One example is that described in U.S. Patent No.

5,896,531 to Curtis et al., in which it is proposed to retain environments that have been initialized to form a pool of environments for future use. Specification, page 1, line 27 – page 2, line 3. In the model proposed by Curtis et al., a second application program's initiation causes the system to examine existing environments for possible reuse if one of them should prove to meet the minimal requirements of the newly-initiated program. Specification, page 2, lines 3-8. Should an examined environment prove not to meet the minimal requirements, it is discarded from the set of existing environments. Specification, page 2, lines 8-10.

The minimal requirements disclosed in Curtis et al. are a set of the parameters required by object-oriented programs for their execution in an object-oriented program system. Specification, page 2, lines 12-15.

In a system as proposed by Curtis et al., a trade-off is found between the saving of initialization time by keeping environments for reuse and the consumption of storage by the system because of the number of pre-initialized environments that are in memory at any time. Specification, page 2, lines 17-21. The trade-off is achieved by the expedient of discarding pre-initialized environments that are found not to fit the second application's minimal requirements. Specification, page 2, lines 22-24.

In such a system, there is an element of wastage, in that environments are discarded which might in time have proven to be useful for subsequent application programs to reuse. Specification, page 2, line 26 – page 3, line 2. Further, the use of environments from the environment pool is limited to those environments which meet the minimal requirements of the second or subsequent application. Specification, page 3, lines 2-5.

Therefore, there is a need in the art to mitigate these problems by finding some more economical way to control the reuse of environments in a system in which

multiple programs and instances of programs may coexist. Specification, page 3, lines 7-10.

The problems outlined above may at least in part be solved in one embodiment of the present invention by a method for progressively improving a fit of a pool of reusable environments to requirements of programs in a computer system. Specification, page 3, lines 14-17. The method may comprise the step of providing a first environment for a first program. Specification, page 3, lines 17-18. The method may further comprise responsive to initiation of a second program, making a determination whether creation of a new environment is a best response. Specification, page 3, lines 19-21. The method may further comprise responsive to a determination that creation of a new environment is a best response, creating a new environment for the second program. Specification, page 3, lines 21-23. The method may further comprise responsive to a determination that creating a new environment is not a best response, testing the pool for a best fit environment. Specification, page 3, lines 23-25. The method may further comprise adding elements to the best fit environment to match requirements of the second program unless the best fit environment already matches the requirements of the second program. Specification, page 3, line 25 – page 4, line 2.

#### VI. ISSUES

Are claims 1-18 properly rejected under 35 U.S.C. §102(b) as being anticipated by Curtis et al. (U.S. Patent No. 5,896,531) (hereinafter "Curtis")?

#### VII. GROUPING OF CLAIMS

Claims 1, 2, 3, 4, 6, 9, 10, 11, 12, 13, 14 and 16 form a first group.

Claims 5 and 15 form a second group.

Claims 7 and 17 form a third group.

Claims 8 and 18 form a fourth group.

The reasons for these groupings are set forth in Appellants' arguments in Section VIII.

#### VIII. ARGUMENT

For a claim to be anticipated under 35 U.S.C. §102, each and every claim limitation must be found within the cited prior art reference and arranged as required by the claim. M.P.E.P. § 2131.

Appellants respectfully assert that Curtis does not disclose "responsive to initiation of a second program, making a determination whether creation of a new environment is a best response" as recited in claim 1 and similarly in claims 9 and 11. The Examiner cites column 5, line 45 – column 6, line 50 and column 7, lines 24-31 of Curtis as disclosing the above-cited claim limitation. Paper No. 7, pages 2 and 5. Appellants respectfully traverse and assert that Curtis instead discloses determining if a task can utilize an existing environment (object management environment). This language is not the same as determining whether to create a new environment. Further, Curtis instead discloses that if the facility does not have an object management environment, then an object management environment reference from the facility name is built and saved for later retrieval. Further, Curtis discloses that after the building and saving of the object management environment reference, a new reusable object management environment is started. However, the new reusable object management environment is started not in response to an initiation of a second program. The language cited in Curtis does not disclose a determination step. That is, the language cited in Curtis does not disclose determining whether to start a new reusable object management environment in response to an initiation of a second program. Furthermore, the language cited in Curtis does not disclose if starting a new reusable object management environment is a best response. Instead, Curtis

discloses that the new reusable object management environment is started if the facility does not have an object management environment. Thus, Curtis does not disclose all of the limitations of claims 1, 9 and 11, and thus Curtis does not anticipate claims 1, 9 and 11. M.P.E.P. §2131.

Appellants further assert that Curtis does not disclose "responsive to a determination that creation of a new environment is a best response, creating a new environment for the second program" as recited in claim 1 and similarly in claims 9 and 11. The Examiner cites column 5, line 45 – column 6, line 50 and column 7, lines 24-31 of Curtis as disclosing the above-cited claim limitation. Paper No. 7, pages 2 and 5. Appellants respectfully traverse and assert that Curtis instead discloses determining if a task can utilize an existing environment (object management environment). This language is not the same as creating a new environment for a second program. Further, Curtis instead discloses that if the facility does not have an object management environment, then an object management environment reference from the facility name is built and saved for later retrieval. Further, Curtis discloses that after the building and saving of the object management environment reference, a new reusable object management environment is started. The language cited in Curtis does not disclose a determination step. That is, the language cited in Curtis does not disclose determining if the creation of a new environment is the best response. The new reusable object management environment in Curtis is started not in response to a determination that the creation of a new environment is the best response. Further, there is no language in Curtis determining a best response in response to the initiation of a program. Hence, Curtis does not disclose creating a new environment for a program in response to determining that the creation of a new environment is the best response in response to the initiation of a program. Instead, Curtis discloses that the new reusable object management environment is started if the facility does not have an object management

environment. Thus, Curtis does not disclose all of the limitations of claims 1, 9 and 11, and thus Curtis does not anticipate claims 1, 9 and 11. M.P.E.P. §2131.

Appellants further assert that Curtis does not disclose "responsive to a determination that creating a new environment is not a best response, testing the pool for a best fit environment" as recited in claim 1 and similarly in claims 9 and 11. The Examiner cites column 5, line 45 – column 6, line 50 of Curtis as disclosing the above-cited claim limitation. Paper No. 7, pages 3 and 5. Appellants respectfully traverse and assert that Curtis instead discloses determining if a task can utilize a reusable object management environment. Curtis further discloses determining if the reusable object management environment is available. The cited passage in Curtis does not disclose determining if a new environment is needed as asserted by the Examiner. Paper No. 7, page 5. The language in the cited passage does not disclose a determination step. There is no language in the cited passage that discloses determining that the creation of a new environment is not the best response in response to the initiation of a program. Hence, the language in the cited passage is not the same as testing a pool of reusable environments for a best fit environment in response to determining that the creation of a new environment is not the best response in response to the initiation of a program. Thus, Curtis does not disclose all of the limitations of claims 1, 9 and 11, and thus Curtis does not anticipate claims 1, 9 and 11. M.P.E.P. §2131.

Appellants further assert that Curtis does not disclose "adding elements to the best fit environment to match requirements of the second program, unless the best fit environment already matches the requirements of the second program" as recited in claim 1 and similarly in claims 9 and 11. The Examiner cites column 5, line 45 – column 6, line 50 and column 8, lines 12-13 of Curtis as disclosing the above-cited claim limitation. Paper No. 7, page 3 and 5. Appellants respectfully traverse and



assert that Curtis instead discloses determining if a task can utilize a reusable object management environment. Curtis further discloses determining if the reusable object management environment is available. Curtis further discloses determining if the available reusable object management environment is incompatible with the task. Curtis further discloses adding a reference to a new reusable object management environment to the list of unnamed object management environments. Adding a reference is not the same as adding elements to an environment where those elements that are added match the requirements of a program. Further, there is no language in Curtis that discloses not adding elements if the environment already matches the requirements of the program. For example, Curtis does not disclose not adding a reference if the environment already matches the requirements of the program. Thus, Curtis does not disclose all of the limitations of claims 1, 9 and 11, and thus Curtis does not anticipate claims 1, 9 and 11. M.P.E.P. §2131.

Appellants further assert that Curtis does not disclose "wherein at least one least recently used of the first, new and best fit environments is eligible to be deleted" as recited in claim 5 and similarly in claim 15. The Examiner cites column 5, line 45 – column 6, line 50 and column 7, line 20 – column 8, line 50 of Curtis as disclosing the above-cited claim limitation. Paper No. 7, pages 3 and 6. Appellants respectfully traverse and assert that Curtis instead discloses that if the object management environment is not reusable or should not be kept or maintained that the object management environment is freed up. However, there is no language in the cited passages that the least recently used object management environment is freed up. Instead, the object management environment is freed up if it is not reusable or should not be kept or maintained. Thus, Curtis does not disclose all of the limitations of claims 5 and 15, and thus Curtis does not anticipate claims 5 and 15. M.P.E.P. §2131.

Appellants further assert that Curtis does not disclose "wherein the step of responsive to initiation of a second program, making a determination whether creation of a new environment is a best response comprises testing whether the pool has reached a maximum size" as recited in claim 7 and similarly in claim 17. The Examiner cites column 7, line 20 – column 8, line 50 and Figure 5 of Curtis as disclosing the above-cited claim limitation. Paper No. 7, pages 4 and 6. Appellants respectfully traverse and assert that Curtis instead discloses that if the pool is not empty then a next reusable object management environment is selected from a list of object management environments. This language is not the same as testing whether the pool has reached a maximum size. Instead, Curtis teaches determining if the pool is not empty. Furthermore, there is no language in the cited passage that discloses a determination step. That is, there is no language in the cited passage that discloses making a determination whether the creation of a new environment is a best response in response to the initiation of a program. Curtis does not disclose all of the limitations of claims 7 and 17, and thus Curtis does not anticipate claims 7 and 17. M.P.E.P. §2131.

Appellants further assert that Curtis does not disclose "wherein the step of responsive to a determination that the pool has reached its maximum size, testing the pool for a best fit environment comprises a programmatically alterable test" as recited in claim 8 and similarly in claim 18. The Examiner cites column 7, line 20 – column 8, line 50 and Figure 5 of Curtis as disclosing the above-cited claim limitation. Paper No. 7, page 4. Appellants respectfully traverse and assert that Curtis instead discloses that if the pool is not empty then a next reusable object management environment is selected from a list of object management environments. As stated above, this language is not the same as testing whether the pool has reached a maximum size. Instead, Curtis teaches determining if the pool is not empty. Further, this language does not disclose testing the pool with a programmatically alterable test. Thus, Curtis

does not disclose all of the limitations of claims 8 and 18, and thus Curtis does not anticipate claims 8 and 18. M.P.E.P. §2131.

As a result of the foregoing, Appellants respectfully assert that not each and every claim limitation was found within the cited prior art reference and thus claims 1-18 are not anticipated by Curtis.

IX. CONCLUSION

For the reasons noted above, the rejections of claims 1-18 are in error. Appellants respectfully request reversal of the rejections and allowance of claims 1-18.

Respectfully submitted,

WINSTEAD SECHREST & MINICK P.C.

Attorneys for Appellants

By: 

Robert A. Voigt, Jr.

Reg. No. 47,159

Kelly K. Kordzik

Reg. No. 36,571

P.O. Box 50784  
Dallas, Texas 75201  
(512) 370-2832

**APPENDIX**

1. A method for progressively improving a fit of a pool of reusable environments to requirements of programs in a computer system, the method comprising steps of:

providing a first environment for a first program;

responsive to initiation of a second program, making a determination whether creation of a new environment is a best response;

responsive to a determination that creation of a new environment is a best response, creating a new environment for the second program;

responsive to a determination that creating a new environment is not a best response, testing the pool for a best fit environment; and

adding elements to the best fit environment to match requirements of the second program, unless the best fit environment already matches the requirements of the second program.

2. The method of claim 1, wherein at least one of the first, new and best fit environments is an execution environment.

3. The method of claim 2, wherein the execution environment is preinitialized.

4. The method of claim 1, wherein at least one of the first, new and best fit environments is eligible to be deleted.

5. The method of claim 4, wherein at least one least recently used of the first, new and best fit environments is eligible to be deleted.

6. The method of claim 1, wherein the elements are parameters of at least one of the first, the new and the best fit environments.

7. The method of claim 1, wherein the step of responsive to initiation of a second program, making a determination whether creation of a new environment is a best response comprises testing whether the pool has reached a maximum size.

8. The method of claim 7, wherein the step of responsive to a determination that the pool has reached its maximum size, testing the pool for a best fit environment comprises a programmatically alterable test.

9. A computer program product, comprising computer program code tangibly embodied in a signal-bearing medium, for, when loaded into a computer system and executed, progressively improving a fit of a pool of reusable environments to requirements of programs in a computer system, by causing the computer system to perform the steps of a method as claimed in claim 1.

10. A computer program product as claimed in claim 9, wherein the signal bearing medium is at least one of a transmissive medium and a storage medium.

11. A computer system for progressively improving a fit of a pool of reusable environments to requirements of programs in a computer system, the computer system comprising:

means for providing a first environment for a first program;

means responsive to initiation of a second program, for making a determination whether creation of a new environment is a best response;

means responsive to a determination that creation of a new environment is a best response, for creating a new environment for the second program;

means responsive to a determination that creating a new environment is not a best response, for testing the pool for a best fit environment; and

means for adding elements to the best fit environment to match requirements of the second program, unless the best fit environment already matches the requirements of the second program.

12. The computer system of claim 11, wherein at least one of the first, new and best fit environments is an execution environment.

13. The computer system of claim 12, wherein the execution environment is preinitialized.

14. The computer system of claim 11, wherein at least one of the first, new and best fit environments is eligible to be deleted.

15. The computer system of claim 14, wherein at least one least recently used of the first, new and best fit environments is eligible to be deleted.

16. The computer system of claim 11, wherein the elements are parameters of at least one of the first, the new and the best fit environments.

17. The computer system of claim 11, wherein the means responsive to initiation of a second program, for making a determination whether creation of a new environment is a best response comprises means for testing whether the pool has reached a maximum size.

18. The computer system of claim 17, wherein the means, responsive to a determination that the pool has reached its maximum size, for testing the pool for a best fit environment comprises means for performing a programmatically alterable test.